

# Universal Kriging Interpolator for Satellite-Derived Global Data

Henry Coakley, Joshua Williams, and Doran Baker

**Abstract**—This paper explores the use of universal Kriging to interpolate sparsely-sampled satellite-based atmospheric data for global display. The background for Kriging is discussed, the base algorithm is developed, the justification for the use of Kriging to interpolate global data is discussed, and ongoing Matlab optimizations are explored.

**Index Terms**—Kriging, radiometry, airglow, data interpolation.

## I. INTRODUCTION

**K**RIGING is a Minimum Mean-Squared Error (MMSE) estimation algorithm. The algorithm generates an unbiased estimator for an unknown location within a stochastic field; this estimator is a linear combination of the given observations of the field. The weighting of the observational data is determined from a statistical model of the desired field, either as a variogram or a correlation model.

*Kriging* derives its name from Daniel Gerhardus Krige, a South African mining engineer who empirically developed this statistical algorithm for calculating distance-weighted average gold grades in South Africa. [1] The theoretical underpinnings of the Kriging algorithm were refined and formalized by Georges Matheron, a French applied mathematician. [2]

Variants of the Kriging algorithm are:

- 1) Simple Kriging
- 2) Ordinary Kriging
- 3) Universal Kriging.

The primary difference between these modifications of the Kriging algorithm is the assumed mathematical model for the mean function of the random field.

The general form of the Kriging estimator is

$$\hat{y}(\mathbf{x}) = \sum_{i=1}^n c_i(\mathbf{x})y(\mathbf{x}_i) + \mu(\mathbf{x})$$

where  $\mathbf{x}$  is a desired location for an estimate of the field,  $y(\mathbf{x}_i)$  are the known values at the locations

$\mathbf{x}_i$ ,  $\mu(\mathbf{x})$  is the mean function of the field, and the  $c_i(\mathbf{x})$  are weighting coefficients which depend upon the location of the estimate, the spatial variance of the model, and the observed data. These coefficients are calculated by minimizing

$$\sigma_k^2 = E[(Y(\mathbf{x}) - \hat{Y}(\mathbf{x}_0))^2]$$

subject to the constraint  $E[Y(\mathbf{x}) - \hat{Y}(\mathbf{x})] = 0$ .

The different variants of the Kriging algorithm arise from the assumption about  $\mu(\mathbf{x}) = E[Y(\mathbf{x})]$ , the mean of the stochastic field:

- 1) Simple Kriging assumes zero mean [ $\mu(\mathbf{x}) = 0$ ]
- 2) Ordinary Kriging implies constant mean [ $\mu(\mathbf{x}) = C$ ]
- 3) Universal Kriging poses a linear regression mean [ $\mu(\mathbf{x}) = \sum_{i=0}^p \beta_i f_i(\mathbf{x})$ ].

## II. KRIGING ALGORITHM

The algorithm development proceeds as follows:

- 1) Appropriate assumptions
- 2) Stochastic field realization
- 3) Unknown parameter estimation
- 4) Solution of regression coefficients.

Much of this derivation is drawn from [3] and [4].

### A. Assumptions

Before beginning the development of the universal Kriging algorithm, it is important to explicitly state the assumptions required. The first assumption is that the observed stochastic field is stationary, i.e., the variance in each spatial dimension is only a function of the spatial difference.

The second set of imposed assumptions is that the locations of the sample points have zero means in each dimension, and the covariance in each dimension is unity. This is achieved by scaling and offsetting the locations before beginning the formulation of the Kriging algorithm.

It is also assumed that there are  $M$  unique observations of the random field.

### B. Stochastic Field Model

The model of the stochastic field is

$$Y(\mathbf{x}) = \mu(\mathbf{x}) + Z(\mathbf{x}), \quad \mu(\mathbf{x}) = \sum_{i=0}^p \beta_i f_i(\mathbf{x}).$$

Note that  $\mu(\mathbf{x})$  is a deterministic function.  $Z(\mathbf{x})$  is a wide-sense stationary zero-mean stochastic field, with

$$E[Z(\mathbf{x})Z(\mathbf{w})] = \sigma^2(\boldsymbol{\theta})\mathcal{R}(\boldsymbol{\theta}, \mathbf{x}, \mathbf{w}),$$

where  $\sigma^2(\boldsymbol{\theta})$  is the process variance and  $\mathcal{R}(\boldsymbol{\theta}, \mathbf{x}, \mathbf{w})$  is the correlation model, both with parameter  $\boldsymbol{\theta}$ . In this approach, it is assumed that there is a deterministic mean function and the variation from the mean can be interpreted as a realization of a stochastic process. The mean function can be written as

$$\mu(\mathbf{x}) = \sum_{i=0}^p \beta_i f_i(\mathbf{x}) = \mathbf{f}(\mathbf{x})^T \boldsymbol{\beta},$$

where  $\boldsymbol{\beta} = [\beta_0 \dots \beta_p]^T$  and  $\mathbf{f}(\mathbf{x}) = [f_0(\mathbf{x}) \dots f_p(\mathbf{x})]^T$ .

Let the matrix  $F$  be defined as

$$F = [\mathbf{f}(\mathbf{x}_1) \dots \mathbf{f}(\mathbf{x}_M)]^T.$$

The correlation matrix  $R$  has elements

$$R_{ij} = \mathcal{R}(\boldsymbol{\theta}, \mathbf{x}_i, \mathbf{x}_j), \quad i, j = 1 \dots M$$

and the cross-correlation for an estimated point is

$$\mathbf{r}(\mathbf{x}) = [\mathcal{R}(\boldsymbol{\theta}, \mathbf{x}, \mathbf{x}_1) \dots \mathcal{R}(\boldsymbol{\theta}, \mathbf{x}, \mathbf{x}_M)]^T.$$

The form of the linear estimator is

$$\hat{y}(\mathbf{x}) = \mathbf{c}^T \mathbf{y}, \quad (1)$$

where  $\mathbf{c} = [c_0 \dots c_p]^T$  and  $\mathbf{y} = [y_1 \dots y_M]^T$ . It follows that

$$\mathbf{y} = F\boldsymbol{\beta} + \mathbf{z}, \quad (2)$$

where  $\mathbf{z} = [z_1 \dots z_M]^T$  are the design site errors.

### C. Parameter Estimation

In order to compute the optimal Kriging estimator, it is necessary to have a numerical representation of the autocorrelation matrix  $R$  and estimates for  $\boldsymbol{\theta}$  and  $\boldsymbol{\beta}$ . Another assumption imposed upon the autocorrelation, in order to make the parameter estimation tractable, is that the autocorrelation is a product of correlations in each dimension, i.e.

$$\mathcal{R}(\boldsymbol{\theta}, \mathbf{x}, \mathbf{w}) = \prod_{j=1}^{\dim(\mathbf{x})} \mathcal{R}_j(\theta_j, x_j, w_j).$$

Common correlation models are members of the exponential family depending on a distance metric between points,

$$\mathcal{R}_j(\theta_j, x_j, w_j) = \exp(-\theta_j |x_j - w_j|^k), \quad k = 1, 2.$$

However, the choice of a correlation model should be governed by the physical phenomena being interpolated.

By [4], the maximum likelihood estimate of  $\boldsymbol{\theta}$  is

$$\hat{\boldsymbol{\theta}} = \arg \min_{\boldsymbol{\theta}} \sigma^2(\boldsymbol{\theta}) |R(\boldsymbol{\theta})|^{\frac{1}{M}}. \quad (3)$$

Since the objective function of  $\boldsymbol{\theta}$  is nonlinear, care must be taken in the minimization algorithm. For a more thorough discussion of this solution, see [4].

There remains the derivation of an estimate for  $\boldsymbol{\beta}$ . Note that by [5], if  $\mathbf{e}$  is a vector of independent, identically distributed random variables and  $\mathbf{x} = A\boldsymbol{\beta} + \mathbf{e}$ , then the best linear unbiased estimator of  $\boldsymbol{\beta}$  is

$$\hat{\boldsymbol{\beta}} = (A^T A)^{-1} A^T \mathbf{x}. \quad (4)$$

However, the random vector  $\mathbf{z}$  is correlated according to the positive definite autocorrelation matrix  $\sigma^2 R = E[\mathbf{z}\mathbf{z}^T]$ . Let  $R = CC^T$  be the Cholesky factorization of  $R$ , and form the vector  $\mathbf{e} = C^{-1}\mathbf{z}$ . Now, the covariance of  $\mathbf{e}$  is

$$\begin{aligned} E[\mathbf{e}\mathbf{e}^T] &= E[(C^{-1}\mathbf{z})(\mathbf{z}^T C^{-T})] \\ &= C^{-1} E[\mathbf{z}\mathbf{z}^T] C^{-T} \\ &= C^{-1} (\sigma^2 R) C^{-T} \\ &= \sigma^2 C^{-1} (CC^T) C^{-T} \\ &= \sigma^2 I. \end{aligned}$$

From equation 5, it can be seen that  $\mathbf{e}$  is a vector of independent random variables. Multiplication of equation 2 by  $C^{-1}$  gives

$$\begin{aligned} C^{-1}\mathbf{y} &= C^{-1}F\boldsymbol{\beta} + C^{-1}\mathbf{z} \\ &= C^{-1}F\boldsymbol{\beta} + \mathbf{e}. \end{aligned}$$

By letting  $A = C^{-1}F$  and  $\mathbf{x} = C^{-1}\mathbf{y}$  in equation 4, the estimate of  $\beta$  becomes

$$\begin{aligned}\hat{\beta} &= ((F^T C^{-T})(C^{-1}F))^{-1} F^T C^{-T} C^{-1} \mathbf{y} \\ &= (F^T R^{-1} F)^{-1} F^T R^{-1} \mathbf{y}.\end{aligned}\quad (5)$$

#### D. MMSE Predictor Coefficient Solution

Having derived all of the necessary prerequisites, the Kriging algorithm can now be developed. The estimation error can be written as

$$\begin{aligned}\epsilon(\mathbf{x}) &= \hat{y}(\mathbf{x}) - y(\mathbf{x}) \\ &= \mathbf{c}^T \mathbf{y} - y(\mathbf{x}) \\ &= \mathbf{c}^T (F\beta + \mathbf{z}) - (\mathbf{f}^T \beta + z) \\ &= \mathbf{c}^T \mathbf{z} - z + (F^T \mathbf{c} - \mathbf{f})^T \beta.\end{aligned}$$

In order to enforce the unbiased requirement on the estimator,

$$\begin{aligned}F^T \mathbf{c} - \mathbf{f} &= 0 \\ \Rightarrow F^T \mathbf{c} &= \mathbf{f}.\end{aligned}\quad (6)$$

The mean squared error (MSE) of this predictor under the previous constraint is

$$\begin{aligned}\varphi(\mathbf{x}) &= E[(\hat{y}(\mathbf{x}) - y(\mathbf{x}))^2] \\ &= E[(\mathbf{c}^T \mathbf{z} - z)^2] \\ &= E[z^2 + \mathbf{c}^T \mathbf{z} \mathbf{z}^T \mathbf{c} - 2\mathbf{c}^T \mathbf{z} z] \\ &= \sigma^2(1 + \mathbf{c}^T R \mathbf{c} - 2\mathbf{c}^T \mathbf{r}).\end{aligned}\quad (7)$$

The next step is to set up a cost function for the constrained optimization,

$$\begin{aligned}L(\mathbf{c}, \boldsymbol{\lambda}) &= \sigma^2(1 + \mathbf{c}^T R \mathbf{c} - 2\mathbf{c}^T \mathbf{r}) \\ &\quad + 2\sigma^2 \boldsymbol{\lambda}^T (F^T \mathbf{c} - \mathbf{f}),\end{aligned}\quad (8)$$

where  $2\sigma^2 \boldsymbol{\lambda}^T$  is a Lagrangian vector for enforcing the constraint in equation 6.

Now, taking the gradient of equation 8 with respect to  $\mathbf{c}$  results in

$$L_c(\mathbf{c}, \boldsymbol{\lambda}) = 2\sigma^2(R\mathbf{c} - \mathbf{r}) - F\boldsymbol{\lambda}.\quad (9)$$

After setting the result of equation 9 equal to zero and solving for  $\mathbf{c}$ ,

$$\mathbf{c} = R^{-1}(\mathbf{r} - F\boldsymbol{\lambda}).\quad (10)$$

Inserting the solution for  $\mathbf{c}$  of equation 10 back into the constraint of equation 6 results in

$$\begin{aligned}F^T(R^{-1}(\mathbf{r} - F\boldsymbol{\lambda})) &= \mathbf{f} \\ \Rightarrow F^T R^{-1} F \boldsymbol{\lambda} &= F^T R^{-1} \mathbf{r} - \mathbf{f} \\ \Rightarrow \boldsymbol{\lambda} &= (F^T R^{-1} F)^{-1} (F^T R^{-1} \mathbf{r} - \mathbf{f}).\end{aligned}\quad (11)$$

Consequently, inserting equation 11 into equation 10 and then substituting this intermediate result for  $\mathbf{c}$  in equation 1 gives

$$\begin{aligned}\hat{y} &= \mathbf{c}^T \mathbf{y} \\ &= (\mathbf{r} - F\boldsymbol{\lambda})^T R^{-1} \mathbf{y} \\ &= \mathbf{r}^T R^{-1} \mathbf{y} - (F^T R^{-1} \mathbf{r} - \mathbf{f}) \\ &\quad (F^T R^{-1} F)^{-1} F^T R^{-1} \mathbf{y}.\end{aligned}\quad (12)$$

By inserting the estimate of  $\beta$  from equation 5, the linear estimator solution from equation 12 becomes

$$\begin{aligned}\hat{y} &= \mathbf{r}^T R^{-1} \mathbf{y} - (F^T R^{-1} \mathbf{r} - \mathbf{f}) \hat{\beta} \\ &= \mathbf{f} \hat{\beta} + \mathbf{r}^T R^{-1} (\mathbf{y} - F \hat{\beta}).\end{aligned}\quad (13)$$

Designating the scaled residual  $R^{-1}(\mathbf{y} - F \hat{\beta})$  in equation 13 as  $\gamma$ , the final form of the linear estimator solution becomes

$$\hat{y}(\mathbf{x}) = \mathbf{f}(\mathbf{x}) \hat{\beta} + \mathbf{r}(\mathbf{x})^T \gamma.\quad (14)$$

Note that  $\gamma$  and  $\hat{\beta}$  in equation 14 depend only on the design points, so these parameters are constant for any particular interpolation point.

It is emphasized that the correlation model and the regression functions are selected by the user. Therefore, once the field parameter estimates  $\hat{\theta}$  and  $\hat{\beta}$  are determined, then the predictor solution is as shown above. The estimated predictor error can be computed by inserting equation 10 into the MSE equation 7.

### III. CURRENT USE FOR VISUALIZATION

One of the major difficulties in visualizing satellite-derived data is the problem of sparse sampling, seen in figure 1. The Kriging algorithm offers an MMSE way to interpolate these sparse data in order to generate high resolution display images for analysis.

Currently, the Kriging toolbox provided by Lophaven et al. [4] is used to generate an intermediate 80x80 point interpolated image, and then a cubic spline interpolation is used to attain the final resolution of 640x480 points [6]. The reason for this choice is because attempting to interpolate the full 640x480 resolution caused Matlab to exit with ‘‘Out of Memory’’ errors. The current test system is an Intel Pentium 4 clocked at 2.4 GHz with 2 GB of DDR-400MHz SDRAM.

#### IV. PROGRAM OPTIMIZATION

For a complete discussion of the original Matlab Kriging toolbox implementation, see [4] and [3]. For a discussion of the hybrid cubic spline interpolation of satellite data, see [6] and [7].

The major optimizations developed so far include:

- 1) Fixing the upper bound on the solution of equation 3
- 2) Reorganizing the predictor generation for memory savings
- 3) Block-processing the full-resolution Kriging image
- 4) Elimination of unnecessary computations and memory usage.

In the original Kriging code, the upper bound for a two dimensional version of equation 3 was  $[20 \ 20]^T$ . However, it was observed that the nonlinear minimization search would hit this bound before finding a global minimizer. Therefore, the bound was increased to  $[90 \ 90]^T$  and an error message was added to inform the user if the bound is encountered before a global minimizer is found.

The original toolbox implementation for determining the predictor from a set of design sites attempted to make full use of the vectorized speed increases that Matlab offers by creating a large matrix of identical columns and then using a term-by-term vector-scalar multiplication. However, when the  $M$  (the number of design sites) becomes large, Matlab depletes all of the available memory attempting to create a replicated-column matrix. The solution to this problem was to replace the vectorized code with an iterative “for” loop. This is less efficient in computation time, but does not result in the depletion of available memory. Therefore, a valid predictor can be generated.

Also, the original toolbox implementation was written to calculate all of the interpolation points in a single call to the function. However, this also leads to memory starvation on Matlab due to the immense number of interpolation points. The solution to this problem was to Krige sub-blocks of the final desired resolution, and then stitch these blocks together into the final image. The reason that this is an acceptable solution is that the predictor structure is independent of the interpolation point locations, so the same interpolator is used to generate an estimate at any point in space.

Finally, the original toolbox always calculated the

MSE estimate, whether or not the user explicitly needed it. By eliminating these computations, the algorithm benefited by a decrease of approximately 10% of the required computation time. Also, the original data set was still present in memory, even after the predictor structure had been generated. By removing unnecessary memory usage, Matlab suffered less “Out of Memory” errors.

#### V. TEST PATTERNS

In figures 2 and 3, the results from the original data visualization method and the optimized method are compared. Each figure contains the true image, the sampled locations from the true image, the interpolated image from the hybrid Kriging algorithm, and the interpolated image from the full-resolution Kriging algorithm.

Figure 2 shows a binary amplitude stripe pattern, using 15-degree longitudinal stripes. Note that the hybrid Kriging tends to smooth the edges of the stripe pattern. The full-resolution Kriging result is much closer to the true image, since it preserves the sharp distinction at the edge of each strip.

Figure 3 shows another binary amplitude stripe pattern, this time implementing 7.5 degree latitudinal strips. Once again, the full-resolution Kriging preserves the detail better than hybrid Kriging.

#### VI. CONCLUSIONS

Full-resolution Kriging clearly offers better performance for simple test patterns. The current optimizations have significantly improved computation times and offer an improved visualization method by using full-resolution Kriging.

Ongoing research topics include:

- 1) How many satellite data-points should be used to generate the predictor?
- 2) Is there a better implementation language than Matlab for some of the computation?
- 3) Is there a better intermediate Kriging resolution for a hybrid approach which offers equivalent performance to full-resolution Kriging, but with faster computation time?

#### ACKNOWLEDGMENTS

The authors would like to thank Dr. Gene Ware for contributions to this research. The authors recognize Richard Fielding, Kenneth Reese, William Harrison, and Brandon Thurgood for previous insight

and development of the Kriging interpolation algorithms. The authors also recognize Jim Russel and Marty Mlynczak for their assistance and insights on SABER data.

## REFERENCES

- [1] S.E. Gano, J.E. Renaud, J.D. Martin and T.W. Simpson, *Update Strategies for Kriging Models for Use in Variable Fidelity Optimization*. 46th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference; Austin, TX; USA; 18-21 April 2005. pp. 1-20
- [2] D. McGrath, C. Zhang, and O.T. Carton, *Geostatistical analyses and hazard assessment on soil lead in Silvermines area*. Ireland. Environ. Poll. 127, 239248.(2004)
- [3] S. Lophaven, H. Nielsen, and J. Sondergaard, *Aspects of the Matlab Toolbox DACE*. Denmark, Sweden: Technical University of Denmark, 2002.
- [4] S. Lophaven, H. Nielsen, and J. Sondergaard, *DACE: A Matlab Kriging Toolbox*, Version 2.0. Denmark, Sweden: Technical University of Denmark, 2002.
- [5] T. Moon and W. Stirling, *Mathematical Methods and Algorithms for Signal Processing*. Upper Saddle River, NJ: Prentice Hall, 2000. pp. 140
- [6] K. Reese, *Processing and Visualization of Mesospheric Hydroxyl Infrared Emission Data from the SABER Sensor*. Masters of Science Thesis: Utah State University, 2005.
- [7] R. Fielding, *Satellite-based Investigation of Mesospheric Infrared Emissions*. Masters of Science Thesis: Utah State University, 2005.
- [8] K. Beaumont, *Homepage: SABER - Sounding of the Atmosphere using Broadband Emission Radiometry*. [Online]. Available: <http://saber.gats-inc.com>. [Accessed Apr. 28, 2008].

**Henry Coakley** Henry Coakley is a Masters of Engineering candidate in the Department of Electrical and Computer Engineering, Utah State University, Logan UT, 84321, USA, email: [hjcoakley@cc.usu.edu](mailto:hjcoakley@cc.usu.edu)

**Joshua Williams** Joshua Williams is a Masters of Science candidate in the Department of Electrical and Computer Engineering, Utah State University, Logan UT, 84321, USA, email: [jwill@cc.usu.edu](mailto:jwill@cc.usu.edu)

**Dr. Doran Baker** Dr. Doran Baker is a Professor of Electrical and Computer Engineering at Utah State University. He serves as a Director of the Rocky Mountain NASA Space Grant Consortium and is a science investigator on the NASA LaRC SABER/TIMED satellite team. Department of Electrical and Computer Engineering, Utah State University, Logan UT, 84321, USA, email: [spacegrant@cc.usu.edu](mailto:spacegrant@cc.usu.edu)

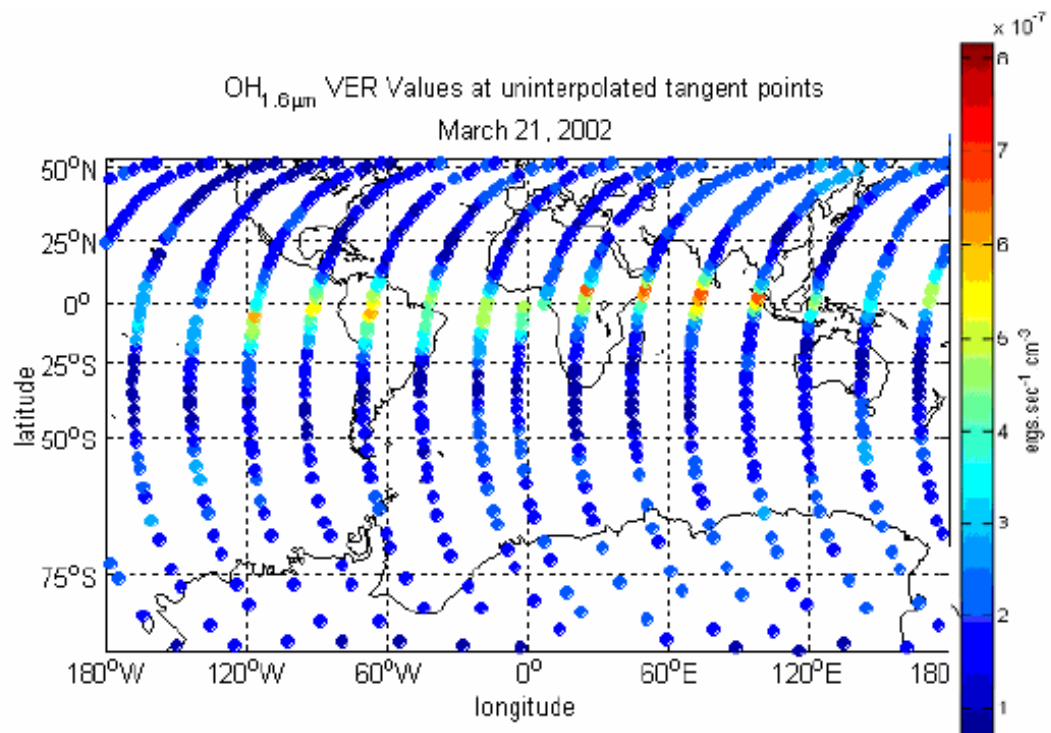


Fig. 1. SABER Uninterpolated Satellite Data. [8]

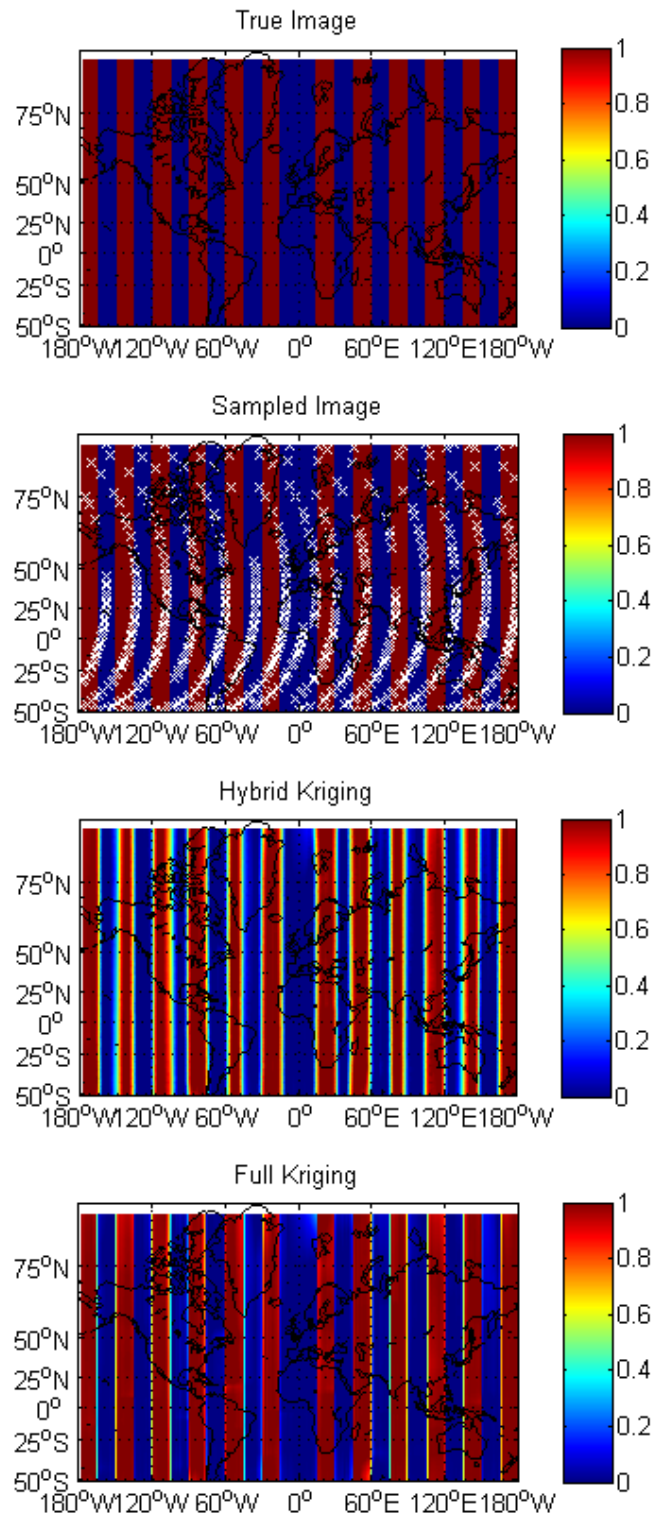


Fig. 2. SABER Interpolated Satellite Data, 15-degree Binary Columns. [8]

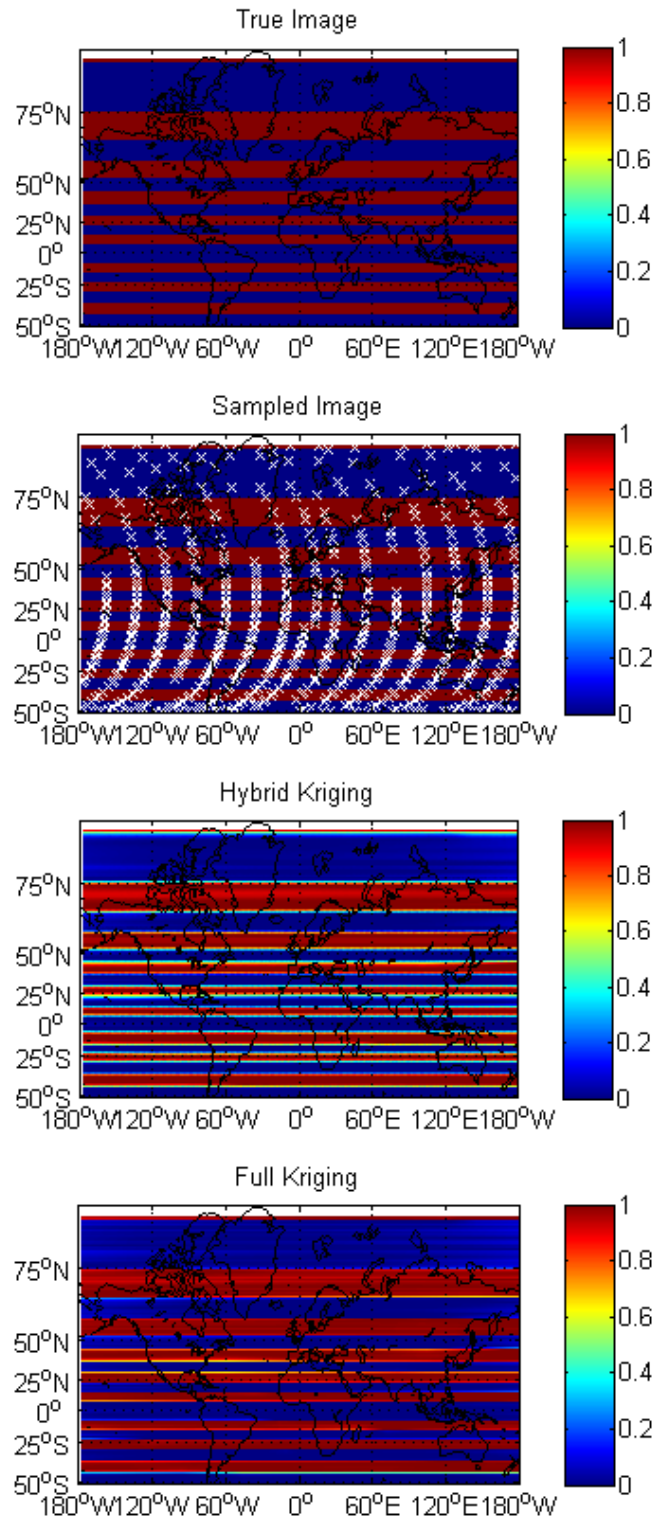


Fig. 3. SABER Interpolated Satellite Data, 7.5-degree Binary Rows. [8]